IW
AF
2183

Our Docket No.: 99-339 / 1496.00059

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Applicant:	Peter Korger		
Application No.:	09/738,485	Examiner:	Harkness, C.
Filed:	December 15, 2000	Art Group:	2183
For:	CONFIGURABLE HARDWARE REGISTER STACK FOR CPU ARCHITECTURES		

CERTIFICATE OF MAILING

I hereby certify that this letter, the response or amendment attached hereto are being deposited with the United States Postal Service as first class mail in an envelope addressed to Mail Stop Appeal Briefs Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on March 17, 2005.

By: Mary Donna Berkley
Mary Donna Berkley

APPEAL BRIEF

Mail Stop - Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant submits the following Appeal Brief pursuant to 37 C.F.R. §41.37 for consideration by the Board of Patent Appeals and Interferences. Please charge $500.00 to cover the cost of filing the opening brief as required by 37 C.F.R. §41.20(b)(2) and any additional fees or credit any overpayment to Deposit Account Number 12-2252.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is the Assignee, LSI Logic Corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to the Appellant, Appellant's legal representative, or Assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-20 are pending and remain rejected. Appellant hereby appeals the rejection of claims 1-20.

IV. STATUS OF AMENDMENTS

Appellant is appealing a final Office Action issued by the Examiner on September 21, 2004. On November 19, 2004, Appellant filed an Amendment After Final requesting reconsideration. On December 14, 2004, the Examiner issued an Advisory maintaining the rejections. On January 17, 2005, Appellant filed a Notice of Appeal based on the last set of claims prior to the Amendment After Final.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A first embodiment of the present invention (as represented by claim 1) concerns a circuit (100) generally comprising a register stack (104) and a control circuit (102). The register stack may be configured as (i) a plurality of segments (124A-124N) addressable through a segment address signal (SEG_ADDR) and (ii) a plurality of registers (126A-126R) within each of the plurality of segments. The plurality of registers may be addressable through a register address signal (REG_ADDR). The register stack is generally described on page 6, lines 5-12 and in FIGS. 1 and 2 of the specification. The control circuit may be configured to (i) store a plurality of register states (in 118), (ii) store a segment count signal (SEG_COUNT) and (iii) present the segment address signal responsive to the plurality of register states, the segment count signal, and the register address signal. The control circuit is generally described on page 6, lines 13-20, page 8, lines 14-21 and FIGS. 1 and 2 of the specification.

A second embodiment of the present invention (as represented by claim 10) concerns a method (FIG. 3) of controlling a register stack. The method generally comprises steps of (A) comparing, (B) gating and (C) addressing. The first step (A) may compare (140) a register address (REG_ADDR) with a plurality of register states (REG_STATES) to present a gating signal (STACK_GATING). Comparing is generally described on page 10, lines 1-2 of the specification. The second step (B) generally gates (142-146) a segment count (SEG_COUNT) with the gating signal to present a segment address (SEG_ADDR). Gating is described on page 10, lines 2-14 of the specification where (i) the signal STACK_GATING in a logical zero state gates (sets) the signal SEG_ADDR to all logical zeros (144) and (ii) the signal STACK_GATING in a logical one state

gates (sets) the signal SEG_ADDR to the signal SEG_COUNT (146). The third step (C) may address (148) the register stack with the register address and the segment address. Addressing is generally described on page 10, lines 14-19 of the specification.

A third embodiment of the present invention (as represented by claim 15) concerns a circuit generally comprising three means for storing and a means for presenting. A first means for storing may store a register stack (104) configured as (i) a plurality of segments (124A-124N) addressable through a segment address (SEG_ADDR) and (ii) a plurality of registers (126A-126R) within each of the plurality of segments. The plurality of registers may be addressable through a register address (REG_ADDR). The first means for storing may be illustrated as the stack 104. The register stack is generally described on page 6, lines 5-12 and in FIGS. 1 and 2 of the specification. A second means for storing may store a plurality of register states (REG_STATES). The second means for storing may be illustrated by the stack control circuit 102, the stack register 105 and/or the status circuit 118. Storing the register states is generally described on page 5, lines 9-15, page 6, lines 13-20 and FIGS. 1 and 2 of the specification. A third means for storing may store a segment count (SEG_COUNT). Storing the segment count is generally illustrated by the counter 120 and/or stack control circuit 102 as described on page 6, lines 17-20 and FIG. 2 of the specification. The means for presenting may present the segment address responsive to the register address and the plurality of register states and the segment count. Presenting the segment address may be illustrated by the stack control circuit 102 and/or the logical AND gate 122 in the specification. The presenting is generally described on page 6, lines 14-20 and FIG. 2 of the specification.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The issue is whether claims 1-20 are patentable under 35 U.S.C. §102(b) over Vegesna et al., U.S. Patent No. 5,226,142 (hereafter Vegesna).

VII. ARGUMENTS

A. 35 U.S.C. §102

The Federal Circuit has stated that "[t]o anticipate, ***every element and limitation*** of the claimed invention must be found in a single prior art reference, ***arranged as in the claim.***"[1] (Emphasis added). The Federal circuit has added that the anticipation determination is viewed from one of ordinary skill in the art: "There must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention."[2] Furthermore, "A claim is anticipated only if each and every element as set forth in the claim is found, either ***expressly or inherently*** described, in a single prior art reference."[3] (Emphasis added).

---

[1] *Brown v. 3M*, 60 USPQ2d 1375, 1376 (Fed. Cir. 2001) citing *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383, 58 USPQ2d 1286, 1291 (Fed. Cir. 2001); *Scripps Clinic & Research Found. v. Genentech Inc.*, 927 F.2d 1565, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991) (Emphasis added by Appellant).

[2] *Scripps Clinic & Research Found. v. Genentech Inc.*, 927 F.2d 1565, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991).

[3] *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, USPQ2d 1051, 1053 (Fed Circ. 1987).

1. Claims 1, 2 and 15 are fully patentable over Vegesna

Claim 1 provides a control circuit configured to store a plurality of register states. Despite the assertion by the Examiner[4], FIG. 6, FIG. 8, column 2, lines 14-25 and column 10, lines 43-64 of Vegesna appear to be silent regarding storing register states in a Result Vector Driver 68 (asserted similar to the claimed control circuit). The cited text of Vegesna reads:

> The typical register file is a memory array, its physical storage locations organized into fixed size windows (i.e. blocks of memory). Each window represents a unique mapping of the thirty-two CPU registers onto an equal number of physical storage locations in the memory array. Each window typically contains twenty-four memory locations which are all allocated to provide physical storage for the local registers. Local registers are used to store local operands. A separate set of eight memory locations is allocated to provide storage for global registers, which are used to store global operands.[5]
>
> Referring to the top portion of FIG. 8, the Result Vector Driver 68 receives two types of inputs over input buses 88 and 92. RESULT Input 88 is a thirty-two bit bus over which a thirty-two bit result of an operation performed by the CPU is input to the Result Vector Driver 68. Select-R input bus 92 is a set of thirty-two selection signals labeled Select-R 92. The Result Vector Driver 68 has a total of 1,024 output bit lines which are input into RAM Array 70. These lines comprise the data inputs for all of the memory locations which are addressable within the currently active register window. The inputs into the RAM Array 70 are labeled according to the four types of registers for which the corresponding memory locations are providing storage: "Global" registers - eight, thirty-two bit vector inputs 91, "Local" registers - eight, thirty-two bit vector inputs 93, "Odd-Result" inputs (for "out" registers if the active window is odd-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 95 and "Even-Result" inputs (for "out" registers if the active window is even-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 97.[6]

Nowhere in the above cited text does Vegesna appear to **expressly** disclose a control circuit configured to store a plurality of register states as presently claimed. FIGS. 6 and 8 of Vegesna do

---

[4] Office Action, November 21, 2004, page 3.

[5] Vegesna, column 2, lines 14-25.

[6] Vegesna. Column 10, lines 43-64.

not appear to cure the text deficiency. Furthermore, the Examiner does not appear to make an **inherency** argument for the Result Vector Driver 68 storing register states. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Assuming, *arguendo*, that the Examiner is arguing for inherency (for which Appellant's representative does not necessarily agree), no evidence or convincing line of reasoning is provided to establish the implied inherency. M.P.E.P. §2112 states:

> "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic <u>necessarily</u> flows from the teachings of the applied prior art. " *Ex parte Levy* 17 USPQ2d 1461, 1464, 1464 (Bd. Pat. App. & Inter. 1990). (Emphasis in original)

Furthermore, inherency requires certainty of results, not mere possibility.[7] In contrast, the Examiner fails to establish by any evidence or convincing line of reasoning (i) that **storing** register states necessarily flows from the Result Vector Driver 68 directing data to the RAM array 70 and (ii) that such storing is a certainty. The assertion by the Examiner[8] that the Result Vector Driver 68 of Vegesna must hold register states in order to know which group of registers to distribute data conflicts with the Examiner's own statement that the Result Vector Driver 68 directs the data "based on the register address."[9] Vegesna also appears to contemplate that the Result Vector Driver 68 is a form of demultiplexer directing data based on a signal SELECT R:

---

[7] See, e.g., *Ethyl Molded Products Co. v. Betts Package, Inc.*, 9 U.S.P.Q. 2d 1001 (E.D.Ky 1988). See also, In re Oelrich, 666 F.2d 578, 581, 212 USPQ 323, 326 (C.C.P.A. 1981).

[8] Advisory, December 14, 2004, page 2.

[9] Office Action, November 21, 2004, page 9, item 26.

> The Result Vector Driver 68 takes incoming results data over thirty-two bit bus RESULT 88 and, in accordance with the one active Select R 92 line out of thirty-two, transmits the data to the appropriate thirty-two bit vector which form the inputs to the one of thirty-two memory locations in the current window of RAM Array 70 which is to have the result stored within it. [10]

Appendix A provides a common definition of a demultiplexer is as follows:

> A *demultiplexer* is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of a specific output line is controlled by the bit values of n selection lines.[11]

The Result Vector Driver 68 of Vegesna could appear to be implemented with 32 demultiplexers, one for each line of a signal RESULT. Since the common demultiplexer defined above does not store data, the Result Vector Driver 68 of Vegesna does not necessarily store data. As such, no inherency exists in the Result Vector Driver 68 for storing a plurality of register states as presently claimed. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Furthermore, the Result Vector Driver 68 of Vegesna does not appear to be aware of a state of the registers 70. In particular, Appendix B provides a common definition of "state" as follows:

> The condition at a particular time of any of numerous elements of computing - a device, a communications channel, a network station, a program, a bit, or other element - used to report on or to control computer operations.[12]

---

[10] Vegesna, column 10 line 64 thru column 11 line 3.

[11] Digital Logic and Computer Design, M. Morris Mano, 1979, pages 172-173.

[12] *Microsoft Computer Dictionary*, Fifth Edition, 2002, definition of "state" and "status"

In contrast, column 10, lines 54-64 of Vegesna refers to the registers has four types, "global", "local" "odd-result" and "even-result". As used in Vegesna, "types" does not appear to match the common definition for "states". Furthermore, no evidence or convincing line of reasoning is provided by the Examiner why one of ordinary skill in the art would consider the register "types" from Vegesna to be similar to similar to the claimed register "states". As such, the assertion by the Examiner[13] that register "states" are functionally equivalent to register "types" appears to be merely a conclusory statement. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Claim 1 further provides that the control circuit is configured to store a segment count signal. Despite the assertion by the Examiner[14], column 3, lines 51-56 and column 3, lines 32-36 of Vegesna appear to be silent regarding the Result Vector Driver 68 (asserted similar to the claimed control circuit) storing a signal CWP (asserted similar to the claimed segment count signal). The cited text of Vegesna reads:

> Only one window of memory locations is active at a time (the window visible to the programmer) and it is identified by a current window pointer (CWP), which is a three bit word stored in the CPU's state register.[15]
>
> In the CY7C601 and CY7C611 the CWP points to a stack of one-hundred twenty-eight, thirty-two bit memory locations. The register file address is an eight bit word which is decoded to select and access one of the register file's one hundred thirty-six memory locations. The CWP comprises the three most significant bits of the register file address. Thus, an increment (or decrement) of the current window pointer offsets the decoded value of the register file address by sixteen. Twenty-four memory locations are accessed for a

---

[13] Office Action, November 21, 2004, page 10, item 32.

[14] Office Action, November 21, 2004, page 3.

[15] Vegesna, column 3, lines 32-36.

> single CWP value, however, thus providing a window overlap of eight memory locations. This overlap in window memory locations creates an effective overlap of window registers which is used to pass parameters from one window to either of its two adjacent windows.[16]

Nowhere in the above cited text does Vegesna appear to mention a control circuit configured to store a segment count signal as presently claimed. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Furthermore, the Examiner appears to be arguing that Vegesna discloses a structure different from as claimed. In particular, the claim provides that the control circuit stores the segment count signal. In contrast, the signal CWP of Vegesna (asserted similar to the claimed segment count signal) appears to have no connection with the Result Vector Driver 68 (asserted similar to the claimed control circuit). Therefore, Vegesna does not appear to disclose or suggest a control circuit configured to store a segment count signal as presently claimed. As such, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Claim 1 further provides that (A) the control circuit is configured to present a segment address signal (B) responsive to (i) the plurality of register states, (ii) a segment count signal and (iii) a register address signal. Despite the assertion by the Examiner[17], (A) the signals on lines 106-112 of Vegesna are not presented from the Result Vector Driver 68 (asserted similar to the claimed control circuit). Furthermore, (B) Vegesna appears to be silent, and no convincing line of reasoning is provided that the signals on lines 106-112 of Vegesna (asserted similar to the claimed segment address signal) are responsive to (i) data stored in the Result Vector Driver 68 (the alleged data

---

[16] Vegesna, column 3, lines 51-56.

[17] Office Action, November 21, 2004, page 3.

asserted to be similar to the claimed register states) and (iii) the signal ADDR R (asserted similar to the claimed register address signal). Therefore, Vegesna does not appear to disclose or suggest a control circuit configured to present a segment address signal responsive to (i) the plurality of register states, (ii) a segment count signal and (iii) a register address signal as presently claimed. As such, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

In summary, the Examiner fails to establish that Vegesna expressly or inherently discloses all of the elements as arranged in the claims. Furthermore, the Examiner makes several conclusory statements not supported by any evidence or convincing line of reasoning why one of ordinary skill in the art would interpret the language of the claims and Vegesna similar to the Examiner. As such, the claimed invention is fully patentable over the cited reference and the rejection should be reversed.

2. Claims 10, 13, 14 and 20 are fully patentable over Vegesna

Claim 10 provides a step for comparing a register address with a plurality of register states to present a gating signal. The Examiner appears to assert[18] that (i) the Result Vector Driver 68 of Vegesna performs the claimed comparing step, (ii) the Result Vector Driver 68 is aware of "resister states" and (iii) the signal ADDR R on line 90 of Vegesna is similar to the claimed register address. Despite the assertion by the Examiner[19], FIG. 8 and column 10, line 43-column 11, line 6

---

[18] Office Action, November 21, 2004, page 6 and page 11, item 36.

[19] OAF, page 3.

of Vegesna appear to be silent regarding the Result Vector Driver 68 comparing the signal ADDR R (asserted similar to the claimed register address) with a plurality of register states allegedly stored within. The cited text of Vegesna reads:

> Referring to the top portion of FIG. 8, the Result Vector Driver 68 receives two types of inputs over input buses 88 and 92. RESULT Input 88 is a thirty-two bit bus over which a thirty-two bit result of an operation performed by the CPU is input to the Result Vector Driver 68. Select-R input bus 92 is a set of thirty-two selection signals labeled Select-R 92. The Result Vector Driver 68 has a total of 1,024 output bit lines which are input into RAM Array 70. These lines comprise the data inputs for all of the memory locations which are addressable within the currently active register window. The inputs into the RAM Array 70 are labeled according to the four types of registers for which the corresponding memory locations are providing storage: "Global" registers - eight, thirty-two bit vector inputs 91, "Local" registers - eight, thirty-two bit vector inputs 93, "Odd-Result" inputs (for "out" registers if the active window is odd-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 95 and "Even-Result" inputs (for "out" registers if the active window is even-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 97. The Result Vector Driver 68 takes incoming results data over thirty-two bit bus RESULT 88 and, in accordance with the one active Select R 92 line out of thirty-two, transmits the data to the appropriate thirty-two bit vector which form the inputs to the one of thirty-two memory locations in the current window of RAM Array 70 which is to have the result stored within it. Store Register Decode Logic Circuit 76 activates the appropriate Select-R signal 92 in accordance with the five bit register address input ADDR-R 90. [20]

Nowhere in the above cited text does Vegesna appear to **expressly** disclose a step for comparing a register address with a plurality of register states to present a gating signal as presently claimed. In addition, FIG. 8 of Vegesna shows that the Result Vector Driver 68 (asserted similar to the claimed control circuit) does not even receive the signal ADDR R (asserted similar to the claimed register address). Furthermore, the Examiner does not appear to make an **inherency** argument for the Result Vector Driver 68 comparing the signal ADDR R to register states stored within. Therefore, *prima*

---

[20] Vegesna. Column 10, line 43-column 11, line 6.

facie anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Assuming, *arguendo*, that the Examiner is arguing for inherency (for which Appellant's representative does not necessarily agree), no evidence or convincing line of reasoning is provided to establish the implied inherency. MPEP §2112 states:

> "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic <u>necessarily</u> flows from the teachings of the applied prior art. " *Ex parte Levy* 17 USPQ2d 1461, 1464, 1464 (Bd. Pat. App. & Inter. 1990)(emphasis in original)

Furthermore, inherency requires certainty of results, not mere possibility.[21] In contrast, the Examiner fails to establish by any evidence or convincing line of reasoning (i) that **storing** register states necessarily flows from the Result Vector Driver 68 directing data to the RAM array 70 and (ii) that such storing is a certainty. The assertion by the Examiner[22] that the Result Vector Driver 68 of Vegesna must hold register states conflicts with the Examiner's own statement that the Result Vector Driver 68 directs the data "based on the register address."[23] Vegesna also appears to contemplate that the Result Vector Driver 68 is a form of demultiplexer directing data based on a signal SELECT R:

> The Result Vector Driver 68 takes incoming results data over thirty-two bit bus RESULT 88 and, in accordance with the one active Select R 92 line out of thirty-two, transmits the data to the appropriate thirty-two bit vector which form the inputs to the one of thirty-two

---

[21] See, e.g., *Ethyl Molded Products Co. v. Betts Package, Inc.*, 9 U.S.P.Q. 2d 1001 (E.D.Ky 1988). See also, In re Oelrich, 666 F.2d 578, 581, 212 USPQ 323, 326 (C.C.P.A. 1981).

[22] Advisory, December 14, 2004, page 6.

[23] Office Action, November 21, 2004, page 9, item 26.

memory locations in the current window of RAM Array 70 which is to have the result stored within it. [24]

Appendix A provides a common definition of a demultiplexer is as follows:

> A *demultiplexer* is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of a specific output line is controlled by the bit values of n selection lines.[25]

The Result Vector Driver 68 of Vegesna could possibly be implemented with 32 demultiplexers, one for each line of a signal RESULT. Since the common demultiplexer defined above does not store data, the Result Vector Driver 68 of Vegesna does not necessarily store data. As such, no inherency exists in the Result Vector Driver 68 for storing a plurality of register states for comparing, as presently claimed. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Furthermore, the Result Vector Driver 68 of Vegesna does not appear to be aware of a state of the registers 70. In particular, Appendix B provides a common definition of "state" as follows:

> The condition at a particular time of any of numerous elements of computing - a device, a communications channel, a network station, a program, a bit, or other element - used to report on or to control computer operations.[26]

In contrast, column 10, lines 54-64 of Vegesna refers to the registers has four types, "global", "local" "odd-result" and "even-result". As used in Vegesna, "types" does not appear to match the common definition for "states". Furthermore, no evidence or convincing line of reasoning is provided by the

[24] Vegesna, column 10 line 64 thru column 11 line 3.

[25] Digital Logic and Computer Design, M. Morris Mano, 1979, pages 172-173.

[26] *Microsoft Computer Dictionary*, Fifth Edition, 2002, definition of "state" and "status"

Examiner why one of ordinary skill in the art would consider the register "types" from Vegesna to be similar to similar to the claimed register "states". As such, the assertion by the Examiner[27] that register "states" are functionally equivalent to register "types" appears to be merely a conclusory statement. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Claim 10 further provides a step for gating a segment count with a gating count signal to present a segment address. The Examiner asserts[28] that (i) signals presented from the Result Vector Driver 68 and (ii) signals on lines 106 -112 combined are similar to the claimed segment address. In contrast, Vegesna appears to consider the signal RESULT as input data, not an address:

> The Result Vector Driver 68 has a total of 1,024 output bit lines which are input into RAM Array 70. These lines comprise the **data inputs** for all of the memory locations which are addressable within the currently active register window.[29] (Emphasis added)

One of ordinary skill in the art would not appear to understand data inputs to a memory as an address. Therefore, the Examiner fails to establish that Vegesna discloses a signal similar to the claimed segment address. As such, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims.

Assuming, *arguendo*, that some unidentified signals presented from the Result Vector Driver 68 of Vegesna are similar to the claimed segment address (for which Appellant's representative does not necessarily agree), Vegesna appears to be silent regarding the signal CWP

---

[27] Office Action, November 21, 2004, page 10, item 32.

[28] Office Action, November 21, 2004, page 6.

[29] Vegesna, column 10, lines 49-54.

(asserted similar to the claimed segment count) being gated by some unidentified signal presented by the Result Vector Driver 68 (asserted similar to the claimed gating signal) to present the signal RESULT from the Result Vector Driver 68. Therefore, Vegesna does not appear to disclose or suggest a step for gating a segment count with a gating count signal to present a segment address s presently claimed.

Assuming, *arguendo*, that signals on lines 106-112 of Vegesna alone are similar to the claimed segment address (for which Appellant's representative does not necessarily agree), Vegesna appears to be silent regarding the signal CWP (asserted similar to the claimed segment count) being gated by some unidentified signal presented from the Result Vector Driver 68 to generate the signals on the lines 106-112 from a Write Decode Logic 78, a Read Decode Logic 80 and a Shared Select Logic 86 of Vegesna. Therefore, Vegesna does not appear to disclose or suggest a step for gating a segment count with a gating count signal to present a segment address as presently claimed.

Furthermore, Vegesna appears to be silent regarding a gating function. Appendix C provides a common definition of "gating" as follows:

> The process of selecting only those portions of a wave between specified time intervals or between specified amplitude limits.[30]

No evidence no convincing line of reasoning is provided by the Examiner for a common gating function in Vegesna. Therefore, *prima facie* anticipation has not been established for lack of evidence of all of the claimed elements as arranged in the claims for lack of evidence that Vegesna discloses a gating step as presently claimed.

---

[30] www.TutorGig.com, Encyclopedia definition 1.

Claim 10 further provides a step for addressing a register stack with the register address and the segment address. In contrast, Vegesna appears to be silent regarding the RAM Array 70 of Vegesna being addressed with the signal ADDR R (asserted similar to the claimed register address). In particular, FIG. 8 shows that the signal ADDR R is decoded by a Store Register Decode Logic 76, the output from which is transferred to the Result Vector Driver 68. Therefore, Vegesna does not appear to disclose or suggest a step for addressing a register stack with the register address and the segment address as presently claimed.

In summary, the Examiner fails to establish that Vegesna expressly or inherently discloses all of the elements as arranged in the claims. Furthermore, the Examiner makes several conclusory statements not supported by any evidence or convincing line of reasoning why one of ordinary skill in the art would interpret the language of the claims and Vegesna similar to the Examiner. As such, the claimed invention is fully patentable over the cited reference and the rejection should be reversed.

3. Claim 3 is fully patentable over Vegesna.

Claim 3 depends from claim 1 and thus contains all of the limitations of claim 1. Consequently, the arguments presented above in support of the patentability of claim 1 are incorporated hereunder in support of claim 3.

Claim 3 further provides at least one of the register states is fixed in a stackable state. Despite the assertion by the Examiner[31], FIG. 6, FIG. 8 and the text in column 2, lines 14-25 of

[31] Office Action, November 21, 2004, page 3.

Vegesna appear to be silent regarding a stackable register state for the RAM Array 70 (asserted similar to the claimed register stack). The cited text of Vegesna reads:

> The typical register file is a memory array, its physical storage locations organized into fixed size windows (i.e. blocks of memory). Each window represents a unique mapping of the thirty-two CPU registers onto an equal number of physical storage locations in the memory array. Each window typically contains twenty-four memory locations which are all allocated to provide physical storage for the local registers. Local registers are used to store local operands. A separate set of eight memory locations is allocated to provide storage for global registers, which are used to store global operands.[32]

Nowhere in the above cited text does Vegesna appear to disclose that the Result Vector Driver 68 (asserted similar to the claimed control circuit storing the register states) holds a fixed stackable state for any portion of the RAM Array 70 (asserted similar to the claimed register array). FIGS. 6 and 8 of Vegesna do not appear to cure the text deficiency. Therefore, Vegesna does not appear to disclose or suggest that at least one of a plurality of register states is fixed in a stackable state as presently claimed. As such, the claim 3 is fully patentable over the cited reference and the rejection should be reversed.

4. Claim 4 is fully patentable over Vegesna.

Claim 4 depends from claim 1 and thus contains all of the limitations of claim 1. Consequently, the arguments presented above in support of the patentability of claim 1 are incorporated hereunder in support of claim 4.

Claim 4 further provides that the register stack further comprises (i) a first portion disposed within a processor and configured as at least one segment of a plurality of segments and

[32] Vegesna, column 2, lines 14-25.

(ii) a second portion disposed external to the processor and configured as at least one segment of the plurality of segments. Despite the assertion by the Examiner[33], FIG. 8 of Vegesna appears to be silent regarding both (i) a processor and (ii) a first portion of the RAM Array 70 (asserted similar to the claimed register stack) being within the processor and a second portion of the RAM Array 70 being external to the processor. Therefore, Vegesna does not appear to disclose or suggest a register stack comprising (i) a first portion disposed within a processor and configured as at least one segment of a plurality of segments and (ii) a second portion disposed external to the processor and configured as at least one segment of the plurality of segments as presently claimed. As such, claim 4 is fully patentable over the cited reference and the rejection should be reversed.

5. Claims 5 and 9 are fully patentable over Vegesna.

Claim 5 depends from claim 1 and thus contains all of the limitations of claim 1. Consequently, the arguments presented above in support of the patentability of claim 1 are incorporated hereunder in support of claim 5.

Claim 9 depends from claim 8 and thus contains all of the limitations of claim 8. Consequently, the arguments presented below in support of the patentability of claim 8 are incorporated hereunder in support of claim 9.

Claim 5 further provides that the control circuit comprises a status circuit configured to present a gating signal responsive to the register address signal. Claim 9 provides language

---

[33] Office Action, November 21, 2004, page 4.

similar to claim 5. Despite the assertion by the Examiner[34], a Store Register Decode Logic 76 of Vegesna (asserted similar to the claimed status circuit) does not appear to form part of the Result Vector Driver 68 (asserted similar to the claimed control circuit). Therefore, Vegesna does not appear to disclose or suggest a control circuit comprising a status circuit configured to present a gating signal responsive to a the register address signal as presently claimed. As such, claims 5 and 9 are fully patentable over the cited reference and the rejection should be withdrawn.

6. Claim 6 is fully patentable over Vegesna.

Claim 6 depends from claim 5 and thus contains all of the limitations of claim 5. Consequently, the arguments presented above in support of the patentability of claim 5 are incorporated hereunder in support of claim 6.

Claim 6 further provides that the status circuit comprises a comparator configured to present a gating signal responsive to a plurality of register states and a register address signal. In contrast, Vegesna appears to be silent regarding any circuitry within the Store Register Decode Logic 76 (asserted similar to the claimed status circuit) that compares a plurality of register states (allegedly stored in the Result Vector Driver 68) with the signal ADDR R (asserted similar to the claimed register address signal). Furthermore, the Examiner fails to argue that Vegesna discloses any circuitry within the Store Register Decode Logic 76 of Vegesna that could potentially be similar to the claimed comparator. Therefore, Vegesna does not appear to disclose or suggest a status circuit comprising a comparator configured to present a gating signal responsive to a plurality of register

---

[34] Office Action, November 21, 2004, page 4.

states and a register address signal as presently claimed. As such, claim 6 is fully patentable over the cited reference and the rejection should be reversed.

7. Claim 7 is fully patentable over Vegesna.

Claim 7 depends from claim 5 and thus contains all of the limitations of claim 5. Consequently, the arguments presented above in support of the patentability of claim 5 are incorporated hereunder in support of claim 7.

Claim 7 further provides that the status circuit comprises a memory device configured to store a plurality of register states and present a gating signal responsive to the plurality of register states and a register address signal. In contrast, Vegesna appears to be silent regarding any circuitry within the Store Register Decode Logic 76 (asserted similar to the claimed status circuit) that stores a plurality of register states (allegedly stored in the Result Vector Driver 68). Furthermore, the Examiner fails to argue that Vegesna discloses any circuitry within the Store Register Decode Logic 76 of Vegesna that could potentially be similar to the claimed memory. Therefore, Vegesna does not appear to disclose or suggest a status circuit comprises a memory device configured to store a plurality of register states and present a gating signal responsive to the plurality of register states and a register address signal as presently claimed. As such, claim 7 is fully patentable over the cited reference and the rejection should be reversed.

8. Claim 17 is fully patentable over Vegesna.

Claim 17 depends from claim 5 and thus contains all of the limitations of claim 5. Consequently, the arguments presented above in support of the patentability of claim 5 are incorporated hereunder in support of claim 17.

Claim 17 further provides that the control circuit comprises (i) (from claim 5) a status circuit configured to present the gating signal in response to the register address signal and (ii) (from claim 17) a plurality of logic gates configured to present the segment address signal responsive to the gating signal and the segment count signal. In contrast, Vegesna appears to be silent regarding any circuitry within the Result Vector Driver 68 (asserted similar to the claimed control circuit) that could be potentially similar to the claimed status circuit and the claimed plurality of logic gates. Furthermore, the Result Vector Driver 68 does not appear to generate the signals on lines 106 and 108 of Vegesna (alleged to carry a signal similar to the claimed segment address signal). Therefore, Vegesna does not appear to disclose or suggest a control circuit comprising (i) a status circuit configured to present a gating signal in response to a register address signal and (ii) a plurality of logic gates configured to present a segment address signal responsive to the gating signal and a segment count signal as presently claimed. As such, claim 17 is fully patentable over the cited reference and the rejection should be reversed.

9. Claim 8 is fully patentable over Vegesna.

Claim 8 depends from claim 17 and thus contains all of the limitations of claim 17. Consequently, the arguments presented above in support of the patentability of claim 17 are incorporated hereunder in support of claim 8.

Claim 8 further provides that the plurality of logic gates are configured to present the segment address signal as a predetermined address responsive to the gating signal having a global state. Despite the assertion by the Examiner[35], Vegesna appears to be silent regarding (i) the Result Vector Driver 68 (asserted similar to the claimed control circuit and allegedly comprising a plurality of logic gates) generating a signal on lines 106 and 108 (alleged to carry a signal similar to the claimed segment address signal) and (ii) an unidentified signal generated by the Result Vector Driver 68 having a global state. Therefore, Vegesna does not appear to disclose or suggest a plurality of logic gates configured to present a segment address signal as a predetermined address responsive to a gating signal having a global state as presently claimed. As such, claim 8 is fully patentable over the cited reference and the rejection should be withdrawn.

10. Claim 16 is fully patentable over Vegesna.

Claim 16 depends from claim 1 and thus contains all of the limitations of claim 1. Consequently, the arguments presented above in support of the patentability of claim 1 are incorporated hereunder in support of claim 16.

---

[35] Office Action, November 21, 2004, page 5.

Claim 16 further provides that the control circuit comprises a counter configured to present the segment count signal. In contrast, Vegesna appears to be silent regarding the Result Vector Driver 68 (asserted similar to the claimed control circuit) includes a counter presenting the signal CWP (asserted similar to the claimed segment count signal). Furthermore, the signal CWP appears to be an input signal, not a signal presented by the Result Vector Driver 68. Therefore, Vegesna does not appear to disclose or suggest a control circuit comprising a counter configured to present a segment count signal as presently claimed. As such, claim 16 is fully patentable over the cited reference and the rejection should be reversed.

11. Claim 11 is fully patentable over Vegesna.

Claim 11 depends from claim 10 and thus contains all of the limitations of claim 10. Consequently, the arguments presented above in support of the patentability of claim 10 are incorporated hereunder in support of claim 11.

Claim 11 further provides steps for (i) presenting a signal communicating a plurality of register states and (ii) selecting one of the plurality of register states as a gating signal based upon a register address. Despite the assertion by the Examiner[36], FIG. 8 and the text in column 10, line 43- column 11, line 6 of Vegesna appear to be silent regarding the claimed steps. The cited text of Vegesna reads:

> The Result Vector Driver 68 has a total of 1,024 output bit lines which are input into RAM Array 70. These lines comprise the data inputs for all of the memory locations which are addressable within the currently active register window. The inputs into the RAM Array 70 are labeled according to the four types of registers for which the corresponding memory

[36]Office Action, November 21, 2004, page 7.

> locations are providing storage: "Global" registers - eight, thirty-two bit vector inputs 91, "Local" registers - eight, thirty-two bit vector inputs 93, "Odd-Result" inputs (for "out" registers if the active window is odd-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 95 and "Even-Result" inputs (for "out" registers if the active window is even-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 97. The Result Vector Driver 68 takes incoming results data over thirty-two bit bus RESULT 88 and, in accordance with the one active Select R 92 line out of thirty-two, transmits the data to the appropriate thirty-two bit vector which form the inputs to the one of thirty-two memory locations in the current window of RAM Array 70 which is to have the result stored within it. Store Register Decode Logic Circuit 76 activates the appropriate Select-R signal 92 in accordance with the five bit register address input ADDR-R 90. [37]

Nowhere in the above cited text does Vegesna appear to mention steps for (i) presenting a signal communicating a plurality of register states and (ii) selecting one of the plurality of register states as a gating signal based upon a register address as presently claimed. FIG. 8 of Vegesna does not appear to cure the text deficiency. As such, *prima facie* anticipation has not been established and the rejection of claim 11 should be reversed.

12. Claim 12 is fully patentable over Vegesna.

Claim 12 depends from claim 10 and thus contains all of the limitations of claim 10. Consequently, the arguments presented above in support of the patentability of claim 10 are incorporated hereunder in support of claim 12.

Claim 12 further provides a step for setting a plurality of register states in response to a reset handler operation. In contrast, the Examiner fails to argue that Vegesna somehow expressly or inherently discloses a reset handler operation or that register states (allegedly stored in

---

[37] Vegesna. Column 10, line 43-column 11, line 6.

the Result Vector Driver 68) are somehow set by the reset handler operation. As such, *prima facie* anticipation has not been established and the rejection of claim 12 should be withdrawn.

13. Claim 18 is fully patentable over Vegesna.

Claim 18 depends from claim 10 and thus contains all of the limitations of claim 10. Consequently, the arguments presented above in support of the patentability of claim 10 are incorporated hereunder in support of claim 18.

Claim 18 further provides a step of presenting a segment address as a predetermined address responsive to a gating signal having a global state. In contrast, Vegesna appears to be silent regarding a signal on lines 106-112 (asserted similar to the claimed segment address signal) being presented in a predetermined state responsive to a gating signal (alleged to be presented by the Result Vector Driver 68) being in a global state. In particular, a Write Decode Logic 78, a Read Decode Logic 80 and a Shared Select Logic 86 of Vegesna that present signals on lines 106-112 do not appear to receive any signals from the Result Vector Driver 68 that could potentially be similar to the claimed gating signal. Therefore, *prima facie* anticipation has not been established and the rejection of claim 18 should be reversed.

14. Claim 19 is fully patentable over Vegesna.

Claim 19 depends from claim 10 and thus contains all of the limitations of claim 10. Consequently, the arguments presented above in support of the patentability of claim 10 are incorporated hereunder in support of claim 19.

Claim 19 further provides a step of storing a register state prior to comparing (from claim 10) with a register address. Despite the assertion by the Examiner[38], the text in column 10, lines 43-64 of Vegesna appear to be silent regarding a sequence between storing register states and comparing register states with other signals. The cited text of Vegesna reads:

> Referring to the top portion of FIG. 8, the Result Vector Driver 68 receives two types of inputs over input buses 88 and 92. RESULT Input 88 is a thirty-two bit bus over which a thirty-two bit result of an operation performed by the CPU is input to the Result Vector Driver 68. Select-R input bus 92 is a set of thirty-two selection signals labeled Select-R 92. The Result Vector Driver 68 has a total of 1,024 output bit lines which are input into RAM Array 70. These lines comprise the data inputs for all of the memory locations which are addressable within the currently active register window. The inputs into the RAM Array 70 are labeled according to the four types of registers for which the corresponding memory locations are providing storage: "Global" registers - eight, thirty-two bit vector inputs 91, "Local" registers - eight, thirty-two bit vector inputs 93, "Odd-Result" inputs (for "out" registers if the active window is odd-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 95 and "Even-Result" inputs (for "out" registers if the active window is even-numbered, otherwise for "in" registers) - eight, thirty-two bit vector inputs 97.[39]

Nowhere in the above cited text does Vegesna appear to mention a step of storing a register state prior to comparing with a register address as presently claimed. As such, *prima facie* anticipation has not been established and the rejection of claim 19 should be reversed.

B. <u>CONCLUSION</u>

Vegesna does not expressly disclose or suggest structure for (i) storing a plurality of register states and (ii) presenting a segment address signal responsive to the register states as presently claimed. Vegesna does not expressly disclose steps for (i) comparing a register address

[38] Office Action, November 21, 2004, page 8.

[39] Vegesna, column 10, lines 43-64.

with a plurality of register states (ii) gating a segment count and (iii) addressing a register stack with a register address as presently claimed. Furthermore, the Examiner fails to establish that the claimed elements/steps are somehow inherent to Vegesna. Hence, the Examiner has clearly erred with respect to the patentability of the claimed invention. It is respectfully requested that the Board overturn the Examiner's rejection of all pending claims, and hold that the claims are not rendered anticipated by the cited reference. However, should the Board find the arguments herein in support of independent claims 1, 11 and/or 15 unpersuasive, the Board is respectfully requested to carefully consider the arguments set forth above in support of each of the independently patentable dependent claims.

Respectfully submitted,

CHRISTOPHER P. MAIORANA, P.C.

Christopher P. Maiorana
Registration No. 42,829

Dated: March 17, 2005

c/o Pete Scott
Intellectual Property Law Department
LSI Logic Corporation
1621 Barber Lane
MS: D-106 Legal
Milpitas, CA 95035

VIII. CLAIM APPENDIX

The claims of the present application which are involved in this appeal are as follows:

1. A circuit comprising:

a register stack configured as (i) a plurality of segments addressable through a segment address signal and (ii) a plurality of registers within each of said plurality of segments, said plurality of registers being addressable through a register address signal; and

a control circuit configured to (i) store a plurality of register states, (ii) store a segment count signal, and (iii) present said segment address signal responsive to said plurality of register states, said segment count signal, and said register address signal.

2. The circuit according to claim 1, wherein at least one of said register states is fixed in a global state

3. The circuit according to claim 1, wherein at least one of said register states is fixed in a stackable state.

4. The circuit according to claim 1, wherein said register stack further comprises:

a first portion disposed within a processor and configured as at least one segment of said plurality of segments; and

a second portion disposed external to said processor and configured as at least one segment of said plurality of segments.

5. The circuit according to claim 1, wherein said control circuit comprises:

a status circuit configured to present a gating signal responsive to said register address signal.

6. The circuit according to claim 5, wherein said status circuit comprises:

a comparator configured to present said gating signal responsive to said plurality of register states and said register address signal.

7. The circuit according to claim 5, wherein said status circuit comprises:

a memory device configured to store said plurality of register states and present said gating signal responsive to said plurality of register states and said register address signal.

8. The circuit according to claim 17, wherein said plurality of logic gates are further configured to present said segment address signal as a predetermined address responsive to said gating signal having a global state.

9. The circuit according to claim 8, wherein said status circuit comprises:

a comparator configured to present said gating signal responsive to said plurality of register states and said register address signal.

10. A method of controlling a register stack comprising the steps of:

(A) comparing a register address with a plurality of register states to present a gating signal;

(B) gating a segment count with said gating signal to present a segment address; and

(C) addressing said register stack with said register address and said segment address.

11. The method according to claim 10, wherein step (A) further comprises the sub-steps of:

presenting a signal communicating said plurality of register states; and

selecting one of said plurality of register states as said gating signal based upon said register address.

12. The method according to claim 10, further comprising the step of:

setting said plurality of register states in response to a reset handler operation.

13. The method according to claim 10, further comprising the step of:

incrementing said segment address in response to a push instruction.

14. The method according to claim 13, further comprising the step of:

decrementing said segment address in response to a pop instruction.

15. A circuit comprising:

means for storing a register stack configured as (i) a plurality of segments addressable through a segment address and (ii) a plurality of registers within each of said plurality of segments, said plurality of registers being addressable through a register address;

means for storing a plurality of register states;

means for storing a segment count; and

means for presenting said segment address responsive to said register address and said plurality of register states and said segment count.

16. The circuit according to claim 1, wherein said control circuit comprises:

a counter configured to present said segment count signal.

17. The circuit according to claim 5, wherein said control circuit further comprises:

a plurality of logic gates configured to present said segment address signal responsive to said gating signal and said segment count signal.

18. The method according to claim 10, further comprising the step of:

presenting said segment address as a predetermined address responsive to said gating signal having a global state.

19. The method according to claim 10, comprises the step of:

storing said register states prior to said comparing.

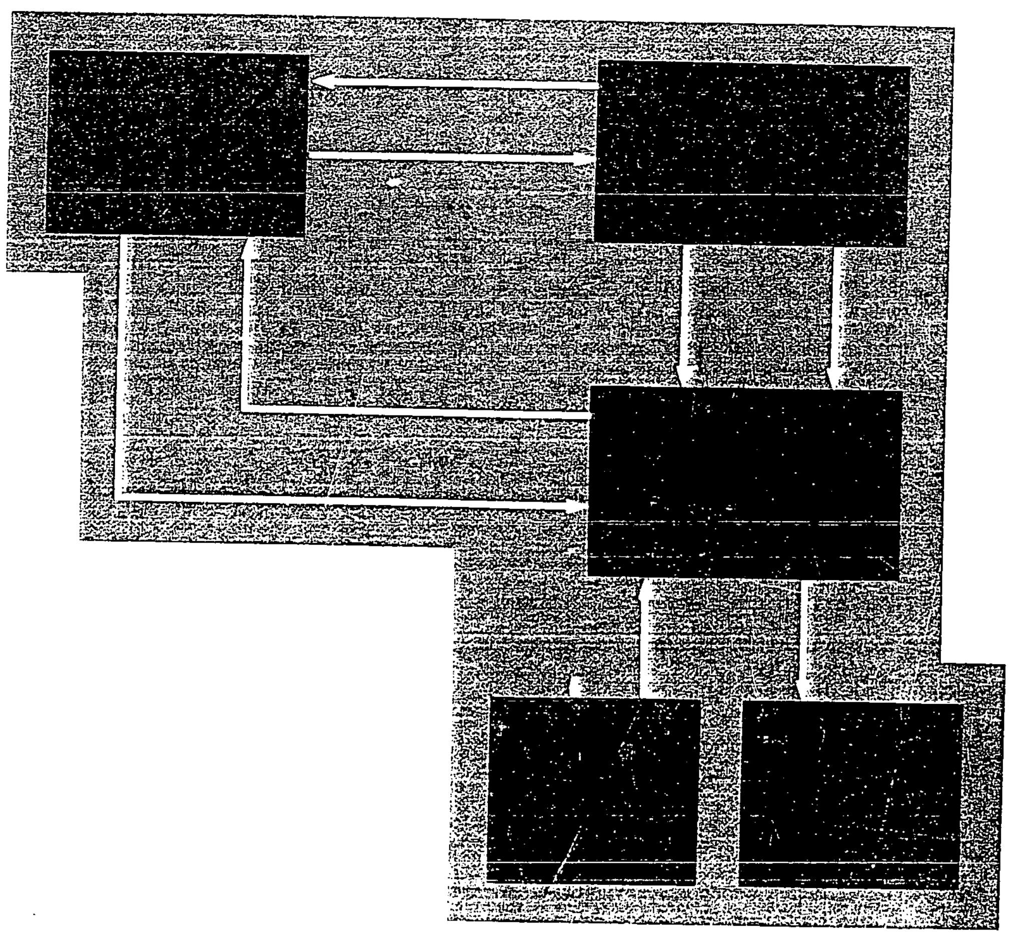
20. The method according to claim 10, further comprising the step of: storing said segment count prior to said gating.

IX. EVIDENCE APPENDIX

Appendix A, Appendix B and Appendix C were made part to the record as attachments to the Amendment After Final of November 19, 2004.

Appendix A

Digital Logic and Computer Design



M. MORRIS MANO

BEST AVAILABLE COPY

Library of Congress Cataloging in Publication Data

Mano, M. Morris (date)
Digital logic and computer design.

Bibliography: p.
Includes index.
1.–Electronic digital computers. 2.–Logic circuits. 3.–Digital integrated circuits. 4.–Logic design. I.–Title.
TK7888.3.M345 621.3815'3 78-21462
ISBN 0-13-214510-3

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Editorial/Production Supervision by Lynn S. Frankel
Cover Design by Edsal Enterprise
Manufacturing Buyer: Gordon Osbourne

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL OF CANADA, LTD., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

complemented form F' with $2^n - k$ minterms. If the number of minterms in a function is greater than $2^n/2$, then F' can be expressed with fewer minterms than required for F. In such a case, it is advantageous to use a NOR gate to sum the minterms of F'. The output of the NOR gate will generate the normal output F.

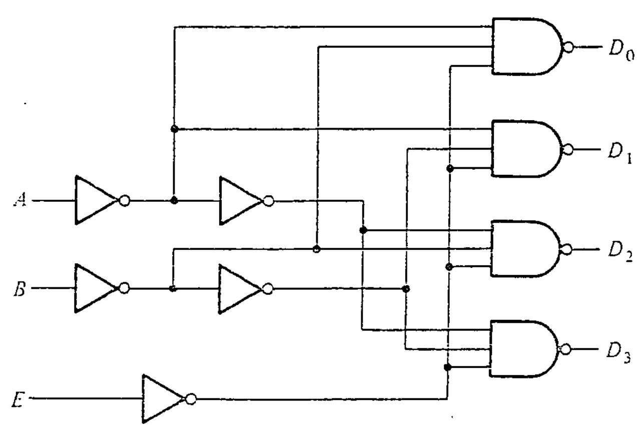
The decoder method can be used to implement any combinational circuit. However, its implementation must be compared with all other possible implementations to determine the best solution. In some cases this method may provide the best implementation, especially if the combinational circuit has many outputs and if each output function (or its complement) is expressed with a small number of minterms.

Demultiplexers

Some IC decoders are constructed with NAND gates. Since a NAND gate produces the AND operation with an inverted output, it becomes more economical to generate the decoder minterms in their complemented form. Most, if not all, IC decoders include one or more *enable* inputs to control the circuit operation. A 2-to-4 line decoder with an enable input constructed with NAND gates is shown in Fig. 5-12. All outputs are equal to 1 if enable input E is 1, regardless of the values of inputs A and B. When the enable input is 0, the circuit operates as a decoder with complemented outputs. The truth table lists these conditions. The X's under A and B are don't-care conditions. Normal decoder operation occurs only with $E = 0$, and the outputs are selected when they are in the 0 state.

The block diagram of the decoder is shown in Fig. 5-13(a). The small circle at input E indicates that the decoder is enabled when $E = 0$. The small circles at the outputs indicate that all outputs are complemented.

A decoder with an enable input can function as a demultiplexer. A *demultiplexer* is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines. The selection of a specific output

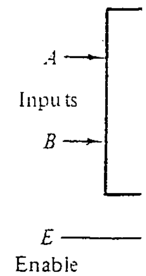


(a) Logic diagram.

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

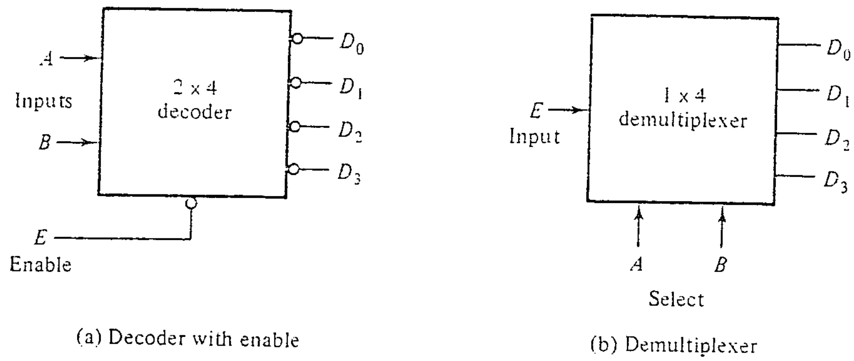
Figure 5-12 A 2-to-4 line decoder with enable (E) input



(a) De

line is controlle
function as a d
and B are take
input variable
directed to only
selection lines
shown in Fig. 5
be the same as
Because decode
a decoder with
enable input th
AND, NAND,
Decoder/
decoder circuit
nected to form

Fi



(a) Decoder with enable

(b) Demultiplexer

Figure 5-13 Block diagrams for the circuit of Fig. 5-12

line is controlled by the bit values of n selection lines. The decoder of Fig. 5-12 can function as a demultiplexer if the E line is taken as a data input line and lines A and B are taken as the selection lines. This is shown in Fig. 5-13(b). The single input variable E has a path to all four outputs, but the input information is directed to only one of the output lines, as specified by the binary value of the two selection lines A and B. This can be verified from the truth table of this circuit, shown in Fig. 5-12(b). For example, if the selection lines $AB = 10$, output D_2 will be the same as the input value E, while all other outputs are maintained at 1. Because decoder and demultiplexer operations are obtained from the same circuit, a decoder with an enable input is referred to as a *decoder/demultiplexer*. It is the enable input that makes the circuit a demultiplexer; the decoder itself can use AND, NAND, or NOR gates.

Decoder/demultiplexer circuits can be connected together to form a larger decoder circuit. Figure 5-14 shows two 3×8 decoders with enable inputs connected to form a 4×16 decoder. When $w = 0$, the top decoder is enabled and the

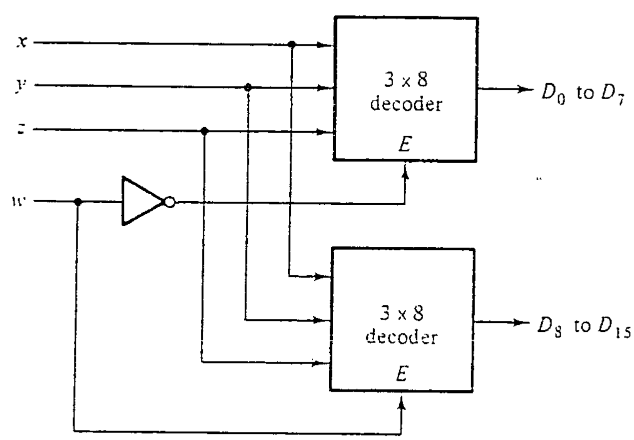
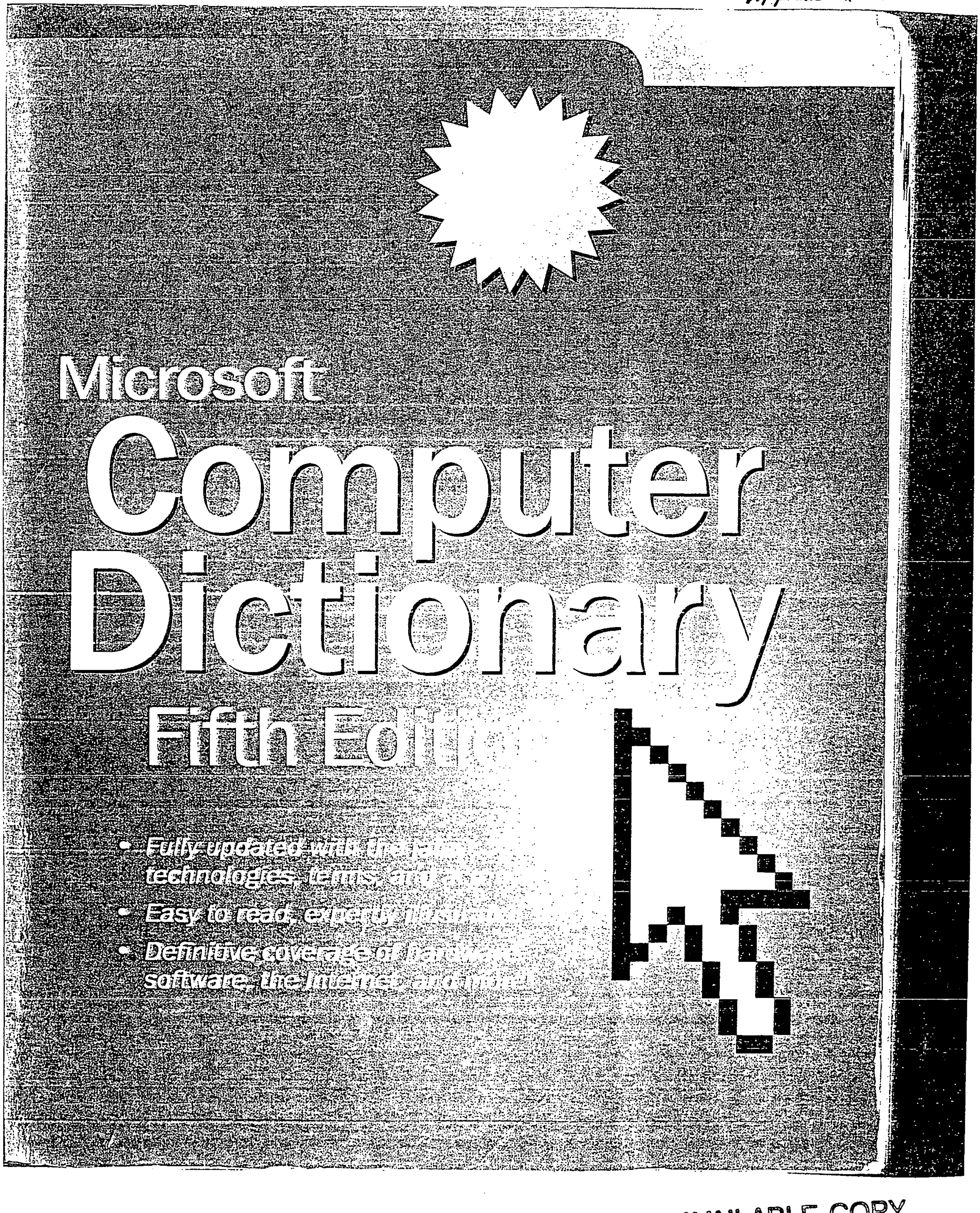


Figure 5-14 A 4×16 decoder constructed with two 3×8 decoders

Appendix B



BEST AVAILABLE COPY

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data
Microsoft Computer Dictionary.--5th ed.
p. cm.
ISBN 0-7356-1495-4
1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

AQ76.5. M52267 2002
004'.03--dc21 200219714

Printed and bound in the United States of America.

2 3 4 5 6 7 8 9 QWT 7 6 5 4 3 2

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com.*

Acquisitions Editor: Alex Blanton
Project Editor: Sandra Haynes

Body Part No. X08-41929

Con

Peter
Bart
Janic
Jeron
Duan
Andre
Robe
Annet
Larry
Thom
Chris
Ilana
Robir
Thom
Willia

Travis
David
Alton

B-2

start/stop transmission *n. See* asynchronous transmission.

startup *n. See* boot^{1}.

startup application *n.* On the Macintosh, the application that takes control of the system when the computer is turned on.

STARTUP.CMD *n.* A special-purpose batch file stored in the root directory of the startup disk in OS/2—the OS/2 equivalent of an MS-DOS AUTOEXEC.BAT file.

startup disk *n. See* system disk.

startup ROM *n.* The bootstrap instructions coded into a computer's ROM (read-only memory) and executed at startup. The startup ROM routines enable a computer to check itself and its devices (such as the keyboard and disk drives), prepare itself for operation, and run a short program to load an operating-system loader program. *See also* boot^{1}, power-on self test.

startup screen *n.* A text or graphics display that appears on the screen when a program is started (run). Startup screens usually contain information about the software's version and often contain a product or corporate logo.

star-wired ring *n.* A network topology in which hubs and nodes connect to a central hub in typical star fashion, but the connections within the central hub form a ring. Star-wired ring is a combination of star and ring topologies.

state *n. See* status.

stateful *adj.* Of or pertaining to a system or process that monitors all details of the state of an activity in which it participates. For example, stateful handling of messages takes account of their content. *Compare* stateless.

stateless *adj.* Of or pertaining to a system or process that participates in an activity without monitoring all details of its state. For example, stateless handling of messages might take account of only their sources and destinations but not their content. *Compare* stateful.

statement *n.* The smallest executable entity within a programming language.

state-of-the-art *adj.* Up to date; at the forefront of current hardware or software technology.

static^{1} *adj.* In information processing, fixed or predetermined. For example, a static memory buffer remains invariant in size throughout program execution. The opposite condition is *dynamic*, or ever-changing.

static^{2} *n.* In communications, a crackling noise caused by electrical interference with a transmitted signal. *See also* noise (definition 2).

static allocation *n.* Apportionment of memory that occurs once, usually when the program starts. The memory remains allocated during the program's execution and is not deallocated until the program is finished. *See also* allocate, deallocate. *Compare* dynamic allocation.

static binding *n.* Binding (converting symbolic addresses in the program to storage-related addresses) that occurs during program compilation or linkage. *Also called:* early binding. *Compare* dynamic binding.

static buffer *n.* A secondary sound buffer that contains an entire sound; these buffers are convenient because the entire sound can be written once to the buffer. *See also* streaming buffer.

static electricity *n.* An electrical charge accumulated in an object. Although generally harmless to humans, the discharge of static electricity through an electronic circuit can cause severe damage to the circuit.

static RAM *n.* A form of semiconductor memory (RAM) based on the logic circuit known as a flip-flop, which retains information as long as there is enough power to run the device. Static RAMs are usually reserved for use in caches. *Acronym:* SRAM. *See also* cache, RAM, synchronous burst static RAM. *Compare* dynamic RAM.

static routing *n.* Routing based on a fixed forwarding path. Unlike dynamic routing, static routing does not adjust to changing network conditions. *Compare* dynamic routing.

static Web page *n.* Web page that displays the same content to all viewers. Usually written in hypertext markup language (HTML), a static Web page displays content that changes only if the HTML code is altered. *See also* dynamic Web page.

station *n.* **1.** In the IEEE 802.11 wireless LAN specification, a single, often mobile, node. **2.** *See* workstation.

stationery^{1} *adj.* Describing a type of document that, when opened by the user, is duplicated by the system; the copy is opened for the user's modification while the original document remains intact. Stationery documents can be used as document templates or boilerplates. *See also* boilerplate, template (definition 5).

stationery^{2} *n.* A stationery document. *See also* stationery^{1}.

S

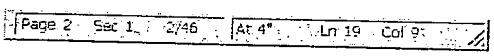
statistical multiplexer *n.* A multiplexing device that adds intelligence to time-division multiplexing by using buffering (temporary storage) and a microprocessor to combine transmission streams into a single signal and to allocate available bandwidth dynamically. *Also called:* stat mux. *See also* dynamic allocation, multiplexing, time-division multiplexing.

statistics *n.* The branch of mathematics that deals with the relationships among groups of measurements and with the relevance of similarities and differences in those relationships. *See also* binomial distribution, Monte Carlo method, probability, regression analysis, standard deviation, stochastic.

stat mux *n. See* statistical multiplexer.

status *n.* The condition at a particular time of any of numerous elements of computing—a device, a communications channel, a network station, a program, a bit, or other element—used to report on or to control computer operations.

status bar *n.* In Windows 9x and Windows NT 4 and later, a space at the bottom of many program windows that contains a short text message about the current condition of the program. Some programs also display an explanation of the currently selected menu command in the status bar. See the illustration.



Status bar.

status codes *n.* Strings of digits or other characters that indicate the success or failure of some attempted action. Status codes were commonly used to report the results of early computer programs, but most software today uses words or graphics. Internet users, especially those with UNIX shell accounts, are likely to encounter status codes while using the Web or FTP. *See also* HTTP status codes.

S

steganography *n.* A "hide-in-plain-sight" technique for concealing information by embedding a message within an innocuous cover message. In steganography, bits of unnecessary data within an image, sound, text, or even a blank file are replaced with bits of invisible information. The term steganography comes from the Greek for "covered writing" and has traditionally included any method of secret communication that conceals the existence of the message. Because steganography cannot be detected by decryption software, it is often used to replace or supplement encryption.

step-frame *n.* The process of capturing video images one frame at a time. This process is used by computers that are too slow to capture analog video images in real time.

stepper motor *n.* A mechanical device that rotates only a fixed distance each time it receives an electrical pulse. A stepper motor is part of a disk drive.

step-rate time *n.* The time required to move a disk actuator arm from one track to the next. *See also* actuator, stepper motor.

stereogram *n. See* autostereogram.

sticky *adj.* In reference to a Web site, properties such as targeted content or services that increase the amount of time users choose to spend at the site and increase user's desire to return to the site repeatedly.

StickyKeys *n.* An accessibility feature built into Macintosh and Windows computers that causes modifier keys such as Shift, Control, or Alt to "stay on" after they are pressed, eliminating the need to press multiple keys simultaneously. This feature facilitates the use of modifier keys by users who are unable to hold down one key while pressing another.

stochastic *adj.* Based on random occurrences. For example, a stochastic model describes a system by taking into account chance events as well as planned events.

stop bit *n.* In asynchronous transmission, a bit that signals the end of a character. In early electromechanical teleprinters, the stop bit provided time for the receiving mechanism to coast back to the idle position and, depending on the mechanism, had a duration of 1, 1.5, or 2 data bits. *See also* asynchronous transmission. *Compare* parity bit, start bit.

Stop error *n.* A serious error that affects the operating system and that could place data at risk. The operating system generates an obvious message, a screen with the Stop error, rather than continuing on and possibly corrupting data. *Also called:* blue screen error, fatal system error. *See also* Blue Screen of Death.

storage *n.* In computing, any device in or on which information can be kept. Microcomputers have two main types of storage: random access memory (RAM) and disk drives and other external storage media. Other types of storage include read-only memory (ROM) and buffers.

storage area network *n.* A high-speed network that provides a direct connection between servers and storage, including shared storage, clusters, and disaster-recovery

devices. A st
nents such as
area networks
"subnetwork"
tion between
on fiber-chan
Mbps and car
mented to pro
required in en
Acronym: SA

storage devi
data in perma
tinction is ma
and secondar
refers to rand
refers to disk

storage loca
item can be fo
uniquely iden
medium.

storage med
on which data
disks, hard di

storage tube

store-and-for
sions in which
mediary befor
and forward is
ets to their de

stored proce
statements an
under a name
an SQL datab
application.

stored progr
scheme, credi
Neumann, in
access storage
allowing code
also von Neun

storefront *n.*

storm *n.* On a
fic. Storms are

STP *n.* Acron
sisting of one

B-4

Appendix C

TutorGig Encyclopedia

Search: gating For Encyclopedia Go

Articles 1 - 14 of 14

Search results for: gating

gating

Encyclopedia Matches : gating

1. Gating
   In telecommunication , the term **gating** has the following meanings b 1. b The process of selecting only those portions of a wave between specified time intervals or between specified amplitude limits. b

2. National Speleological Society
   The National Speleological Society NSS is an organization formed to advance the exploration, conservation, study, and understanding of caves . Members may engage in mapping, cleaning, gating sensitive ...

3. Intensified charge-coupled device
   of **gating** the MCP also offers the possibility to gate ICCD cameras very fast. Therefore ICCD cameras

4. Frog (disambiguation)
   term referring to a French person. FROG stands for Frequency Resolved Optical **Gating**, a method

5. Significant condition
   , or **gating** . Source from Federal Standard 1037C and from MIL STD 188

6. Flow cytometry
   extractions which are termed gates. Specific **gating** protocols exist for diagnostic and clinical purposes

7. Time-resolved spectroscopy
   gating femtoseconds nanoseconds a short laser pulse acts as a gate for the detection of fluorescence

8. Connex
   with indequate electric power supplies, for **gating** and fencing stations to reduce freeloader

9. Proctor
   as **gating** . They have to draw up the list of candidates for examination, and have to be present at all ... number of days **gating** . In the case of more serious offences the proctor generally reports the matter

10. G protein
    molecule itself, by activating other second messengers or by **gating** ion channel s directly. For example

11. Nuclear medicine
    often called a dynamic dataset, a cardiac gated cardiac **gating** time sequence, or a spatial sequence

12. Classic RISC pipeline
    , generally by **gating** off the clock to the flip flops at the start of each stage. The disadvantage of this strategy

13. Personal rapid transit
    the vehicle, the more costly the track, and the track is the **gating** system cost. As well, large tracks

14. List of electronics topics

Attorney Docket: 1496.00059

IN RE APPLICATION OF: Peter Korger

SERIAL NO.: 09/738,485

TITLE: CONFIGURABLE HARDWARE REGISTER STACK FOR CPU ARCHITECTURES

FILED: December 15, 2000

EXAMINER: Harkness, C.

ART UNIT: 2183

RESPONSE TRANSMITTAL AND EXTENSION OF TIME REQUEST (IF REQUIRED)

OIPE MAR 2 1 2005 PATENT & TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

FEE CALCULATION FOR ENCLOSED AND EXTENSION REQUEST (IF ANY)

	Claims Remaining	Highest No. Previous	Extra Rate	Additional Fee
Total Claims	20 minus	20 =	0 x $ 50.00	$ 0.00
Independent Claims	3 minus	3 =	0 x $200.00	$ 0.00
Multiple Dependent Claim First Added			+ $360.00	$ 0.00

TOTAL IF NOT SMALL ENTITY . . $0.00

[ ] SMALL ENTITY STATUS - If applicable, divide by 2 . . . . $0.00

[ ] Applicant also requests a ___ month extension of time for response to the outstanding Office Action. The fee is $0.00

[X] Fee set forth for Filing of Appeal Brief $500.00

TOTAL FEE $500.00

The Commissioner is hereby authorized to charge all fees associated with this Communication to Deposit Account No. 12-2252.

CHRISTOPHER P. MAIORANA, P.C.

By: ______________________
Christopher P. Maiorana
Registration No.: 42,829

24840 Harper Avenue, Suite 100
St. Clair Shores, Michigan 48080
(586) 498-0670

I hereby certify that this letter, the response or amendment attached hereto are being deposited with the United States Postal Service as first class mail in an envelope addressed to Mail Stop - Appeal Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on March 17, 2005.

By: Mary Donna Berkley
Mary Donna Berkley